Claims

1. Method for controlling functional units in a processor, according to which, in a configuration, a sequence of primary instruction words consisting of multiple instruction word parts and originating from a translation of a program code is compressed and stored as a sequence of related program words, and according to which, in a subsequent execution phase, sequential secondary instruction words consisting of a plurality of instruction word parts and having the full instruction word width needed to control all functional units are generated from the sequence of program words **characterized in that**,

   in the result of the configuration, a program word (7) has a first characteristic (10) of a primary instruction word (5) from a first group (11) of preceding primary instruction words (5), which has the greatest similarity to the primary instruction word (5) associated with the program word (7), and contains instruction word parts (6) that differentiate the primary instruction word (5) belonging to the program word (7) from the primary instruction word (5) belonging to the first characteristic (10)

   and in that, in the execution phase, a second group (12) of secondary instruction words (9) — corresponding in number to the first group (11) — each of which is provided with a second characteristic (13), is stored, and in that, by means of the first characteristic (10) contained in the program word (7), a secondary instruction word (9) corresponding to the associated primary instruction word (5) is ascertained from the second group (12) via the associated second characteristic (13), and the secondary instruction word (9) corresponding to the program word (7) is generated in that the instruction word parts (6) contained in the program word (7) are exchanged in the secondary instruction word (9) from the second group (12).

2. Method in accordance with claim 1, **characterized in that** the first group (11) consists of a first number of primary instruction words (5) that directly precede the primary instruction word (5) in question and

in that the second group (12) consists of a second number of secondary instruction words (9) that is at least equal to the first number, where, prior to the generation of the next sequential secondary instruction word (9), each most recent secondary instruction word (9) is appended to the second group (12) as the last word, and the first secondary instruction word (9) to have been added and that is in excess of the second number is removed from the second group (12).

3. Method in accordance with claim 1, **characterized in that** the newly generated secondary instruction word (9) is appended to the second

group (12) in that the former is stored in place of the secondary instruction word (9) that was used for its generation.

4. Method in accordance with claim 1, **characterized in that** the newly generated secondary instruction word (9) is not stored.

5. Method in accordance with one of claims 1 through 4, **characterized in that** the first characteristic (10) is formed as a minimum code distance between the primary instruction word belonging to the program word in question and the primary instruction word with the greatest similarity.

6. Method in accordance with one of claims 1 through 5, **characterized in that** the second characteristic (13) consists of an address corresponding to the first characteristic that is the address of a preceding secondary instruction word in a memory (14) used for storage of the second group (12).

7. Method in accordance with one of claims 1 through 6, **characterized in that** the program word (7) consists of a number of instruction word parts (6) that corresponds to the number of instruction word parts (6) to be differentiated that occurs most frequently within the configuration, and in that a plurality of program words (7) are used to assemble secondary instruction words (9) that require more than the number of instruction

words (9) stored in one program word (7) for the secondary instruction word (9) used for generation.

8. Method in accordance with one of claims 1 through 7, **characterized in that** the instruction word parts (6) are compressed in one program word (7) by reducing the bit width to the extent that it is possible to represent the most frequently occurring instruction word parts (6), and in that multiple program words (7) are used when instruction word parts (6) occur that require a greater bit width in order to be represented.

9. Method in accordance with claim 8, **characterized in that** the width of the instruction word parts (6) in the program word (7) is halved, and one or two program words (7) are provided for representation of the instruction word parts (6).

10. Processor arrangement for carrying out the method, having functional units, an instruction word memory associated with these functional units and an instruction word buffer for storing instruction words that have already been generated and have a width that is at least the size of the bit width of the secondary instruction word, **characterized in that** the instruction word buffer consists of a memory (14) with selective line-by-line access.